

VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLL	IIIIIIII	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLL	IIIIIIII	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLL	IIIIIIII	0000000000	
VVV		VVV	MMMMMM	MMMMMM	SSS	LLL	III	000	000
VVV		VVV	MMMMMM	MMMMMM	SSS	LLL	III	000	000
VVV		VVV	MMMMMM	MMMMMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSSSSSSSSS	LLL	III	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSS	LLL	III	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSS	LLL	III	0000000000	
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSS	LLL	III	000	000
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLLLLLLLLLLLLLLL	IIIIIIII	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLLLLLLLLLLLLLLL	IIIIIIII	0000000000	
VVV		VVV	MMM	MMM	SSSSSSSSSSSS	LLLLLLLLLLLLLLLL	IIIIIIII	0000000000	

```
SSSSSSSS TTTTTTTTTT AAAAAA RRRRRRRR MM MM IIIIII SSSSSSSS CCCCCCCC
SSSSSSSS TTTTTTTTTT AAAAAA RRRRRRRR MM MM IIIIII SSSSSSSS CCCCCCCC
SS SS TT AA AA RR RR MMMM MMMM II II SS SS CC
SS SS TT AA AA RR RR MMMM MMMM II II SS SS CC
SS SSSSSS TT AA AA RRRRRRRR MM MM II II SS SS CC
SS SSSSSS TT AA AA RRRRRRRR MM MM II II SS SS CC
SS SS TT AA AA RRRRRRRR RR RR MM MM II II SS SS CC
SS SS TT AA AA RR RR RR RR MM MM II II SS SS CC
SSSSSSSS TT AA AA RR RR RR RR MM MM II II SS SS CC
SSSSSSSS TT AA AA RR RR RR RR MM MM IIIIII SSSSSSSS CCCCCCCC
SSSSSSSS TT AA AA RR RR RR RR MM MM IIIIII SSSSSSSS CCCCCCCC

MM MM AAAAAA RRRRRRRR
MM MM AAAAAA RRRRRRRR
MMM MMM AA AA RR RR
MMM MMM AA AA RR RR
MM MM AA AA RR RR
MM MM AA AA RRRRRRRR
MM MM AA AA RRRRRRRR
MM MM AAAAAA RR RR
MM MM AAAAAA RR RR
MM MM AA AA RR RR
MM MM AA AA RR RR
MM MM AA AA RR RR
MM MM AA AA RR RR
```

.TITLE STARMISC - MISCELLANEOUS SYSTEM SERVICE MACROS
.IDENT 'V04-000'

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

++
FACILITY: System Service Macros

ABSTRACT:

This module contains some miscellaneous macros used by the system
services.

ENVIRONMENT:

AUTHOR: Various VMS developers, CREATION DATE: 26-Aug-1982

MODIFIED BY:

V03-002 CWH0002 CW Hobbs 30-Apr-1983
Move \$INPUT and \$OUTPUT from STARLET.SDL to this file
as these macros are not suitable for general language
use.

V03-001 CWH0001 CW Hobbs 26-AUG-1982
Edit STARLET.MAR into STARMISC.MAR as part of the
conversion to STARLET.SDL. This contains various
macros invoked by the system service definitions.

V02-030 KTA0081 Kerbey T. Altmann 22-Feb-1982
Add \$MOVEADR macro

--

The following macros are used with system service macro definitions to generate the correct addressing, or to optimize adjacent parameters which are often defaulted.

PUSH ADDRESS MACRO

This macro generates a push address instruction with the correct context. If the address argument was defaulted, a zero is pushed on the stack.

```
.MACRO $PUSHADR,ADDR,CONTEXT=L
    .IF      IDN,0,ADDR
    PUSHL    #0
    .IFF
    PUSHA'CONTEXT    ADDR
    .ENDC
.ENDM $PUSHADR
```

MOVE ADDRESS MACRO

This macro generates a move address instruction with the correct context. If the address argument was defaulted, the destination is cleared.

```
.MACRO $MOVEADR,ADDR,DST,CONTEXT=L
    .IF      IDN,0,ADDR
    CLR'CONTEXT    DST
    .IFF
    MOVA'CONTEXT    ADDR,DST
    .ENDC
.ENDM $MOVEADR
```

MACRO TO PUSH QUADWORD FOR TWO DEFAULTED VALUE ARGUMENTS

```
.MACRO $PUSHTWO A,B
    $ST1 = 0
    .IF IDN,<#0>,<A>
    .IF IDN,<#0>,<B>
    $ST1 = 1
    .ENDC
    .ENDC
    .IF NE $ST1
    CLRQ -(SP)
    .IFF
    PUSHL A
    PUSHL B
    .ENDC
.ENDM $PUSHTWO
```

MACRO TO CHECK QIO ARGUMENTS FOR POSSIBLE QUAD PUSH.
THE FIRST ARG IS A VALUE AND THE SECOND IS AN ADDRESS.

```
.MACRO $QIOPUSH VAL,ADR
    $ST1 = 0
    .IF IDN,<#0>,<VAL>
    .IF IDN,<0>,<ADR>
    $ST1 = 1
    .ENDC
    .ENDC
    .IF NE $ST1
    CLRQ -(SP)
    .IFF
    PUSHL VAL
    $PUSHADR ADR
    .ENDC
.ENDM $QIOPUSH
```

MACRO TO CHECK FOR QUAD CLEAR IN \$ASSIGN. FIRST ARGUMENT IS
A QUADWORD ADDRESS, SECOND IS A VALUE.

```
.MACRO $ASNPUSH ADRQ,VAL
    $ST1 = 0
    .IF IDN,<0>,<ADRQ>
    .IF IDN,<#0>,<VAL>
    $ST1 = 1
    .ENDC
    .ENDC
    .IF NE $ST1
    CLRQ -(SP)
    .IFF
    $PUSHADR ADRQ,CONTEXT=0
```

STARMISC.MAR;1

16-SEP-1984 17:07:52.70 Page 4

.ENDM PUSHL VAL
 .ENDC
 \$ASNPUSH

```

: MACRO TO GENERATE GENERAL ARGUMENT LIST.
: THE FIRST PARAMETER IS THE TOTAL NUMBER OF ARGUMENTS TO GENERATE.
: ANY DEFAULTED ARGUMENTS GENERATE A LONGWORD OF ZERO.
:

```

```

.MACRO $ARGLIST LEN,P1,P2,P3,P4,P5,P6,P7,P8,P9,PA,PB,PC,PD,-
    PE,PF
    $ST1=0
    .ADDRESS          LEN
    .IRP $ST2,<P1,P2,P3,P4,P5,P6,P7,P8,P9,PA,PB,PC,PD,PE,PF>
    .IF EQ LEN-$ST1
    .MEXIT
    .ENDC
    .IF NB $ST2
    .ADDRESS          $ST2
    .IFF
    .ADDRESS          0
    .ENDC
    $ST1=$ST1+1
    .ENDM
.ENDM $ARGLIST

```

```

: MACRO TO GENERATE OFFSET DEFINITION NAMES. THE OFFSETS ARE DEFINED
: WITHIN THE $NAME FORM OF THE MACRO.
:

```

```

.MACRO $OFFDEF,MNAME,LIST
    .NLIST
    $ST1 = 4
    $$ARGS=0
    .IRP $ST2,<LIST>
    MNAME'$ '$ST2 = $ST1
    $ST1 = $ST1 + 4
    $$ARGS=$$ARGS+1
    .ENDM
    MNAME'$ _NARGS = $$ARGS
    .LIST
.ENDM $OFFDEF

```

```

:++
: MACRO TO GENERATE SYSTEM BUGCHECK
:--

```

```

.MACRO $BUG_CHECK ERROR,TYPE=CONT
    .WORD ^XFFFF
    .IIF IDN <TYPE>,<FATAL> , .ADDRESS <'ERROR'&^X0FFFFFFF8>!4
    .IIF DIF <TYPE>,<FATAL> , .ADDRESS 'ERROR'
.ENDM $BUG_CHECK

```

```

:++
: $INPUT
:
: $INPUT Macro
:
: $INPUT chan ,length ,buffer ,[iosb] ,[efn]
:

```



```

chan = number of the channel on which I/O is to be performed
length = length of the input buffer
buffer = address of the input buffer
iosb = address of quadword I/O status block
efn = event flag to set on completion

```

```

.MACRO $INPUT CHAN,LENGTH,BUFFER,IOSB=0,EFN=#0
  $$QIOINPUT INPUT,CHAN,LENGTH,BUFFER,IOSB,EFN
.ENDM $INPUT

```

```

++ $OUTPUT

```

```

$OUTPUT Macro

```

```

$OUTPUT chan, length, buffer, [iosb], [efn]

```

```

chan = channel on which I/O is directed
length = length of the output buffer
buffer = address of the output buffer
iosb = address of quadword I/O status block
efn = event flag number to set on completion

```

```

.MACRO $OUTPUT CHAN,LENGTH,BUFFER,IOSB=0,EFN=#0
  $$QIOOUTPUT OUTPUT,CHAN,LENGTH,BUFFER,IOSB,EFN
.ENDM $OUTPUT

```

```

++ MACRO TO CONVERT $INPUT CALL TO $QIOW CALL

```

```

.MACRO $$QIOINPUT QIOTYPE,QIOCHAN,QIOLENGTH,QIOBUFFER,QIOIOSB,QIOEFN
  $IODEF
  $QIOW_S EFN=<QIOEFN>,CHAN=<QIOCHAN>,FUNC=#IOS_READVBLK,IOSB=<QIOIOSB>,-
  P1=<QIOBUFFER>,P2=<QIOLENGTH>
.ENDM $$QIOINPUT

```

```

++ MACRO TO CONVERT $OUTPUT CALL TO $QIOW CALL

```

```

.MACRO $$QIOOUTPUT QIOTYPE,QIOCHAN,QIOLENGTH,QIOBUFFER,QIOIOSB,QIOEFN
  $IODEF
  $QIOW_S EFN=<QIOEFN>,CHAN=<QIOCHAN>,FUNC=#IOS_WRITEVBLK,IOSB=<QIOIOSB>,-
  P1=<QIOBUFFER>,P2=<QIOLENGTH>,P4=#32
.ENDM $$QIOOUTPUT

.LIST

```


0434 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY